

Εργαστήριο 10 – Άσκηση 1 - Ανάλυση

Εκφώνηση: Στο moodle σας δίνονται οι εικόνες έξι ηρώων της ταινίας Star Wars. Για κάθε ήρωα σας δίνεται μία μεγάλη και μία μικρή εικόνα. Επίσης, σας δίνονται και οι εικόνες ενός δεξιού και ενός αριστερού βέλους. Στην εκκίνηση η εμφάνιση της εφαρμογής θα πρέπει να είναι η παρακάτω:



Στόχος της άσκησης είναι, οι εικόνες των έξι ηρώων να μπουν σε μία σειρά. Κάθε φορά που ο χρήστης επιλέγει το δεξί βελάκι, να γυρίζουν δεξιόστροφα, ενώ όποτε επιλέγει το αριστερό βελάκι, αριστερόστροφα. Όταν, θα φτάνει στο τέλος της σειράς θα ξεκινά από την αρχή. Κάθε φορά από τις τρεις μικρές εικόνες που θα εμφανίζονται, η μεσαία εικόνα θα παρουσιάζεται και στη μεγάλη της διάσταση.

Ανάλυση και εξήγηση λύσης

Βήμα 1: Δημιουργούμε ένα νέο project στο NetBeans με όνομα Askisi12.

Βήμα 2: Δημιουργούμε ένα νέο Java αρχείο τύπου JFrame Form με όνομα Ergastirio10_1.

Βήμα 3: Ορισμός Ιδιοτήτων στο JFrame. Ορίζουμε ως τίτλο του JFrame το «Star Wars Carousel».

Βήμα 4: Κατεβάζουμε τις εικόνες από το moodle και τις αποθηκεύουμε στο Project.

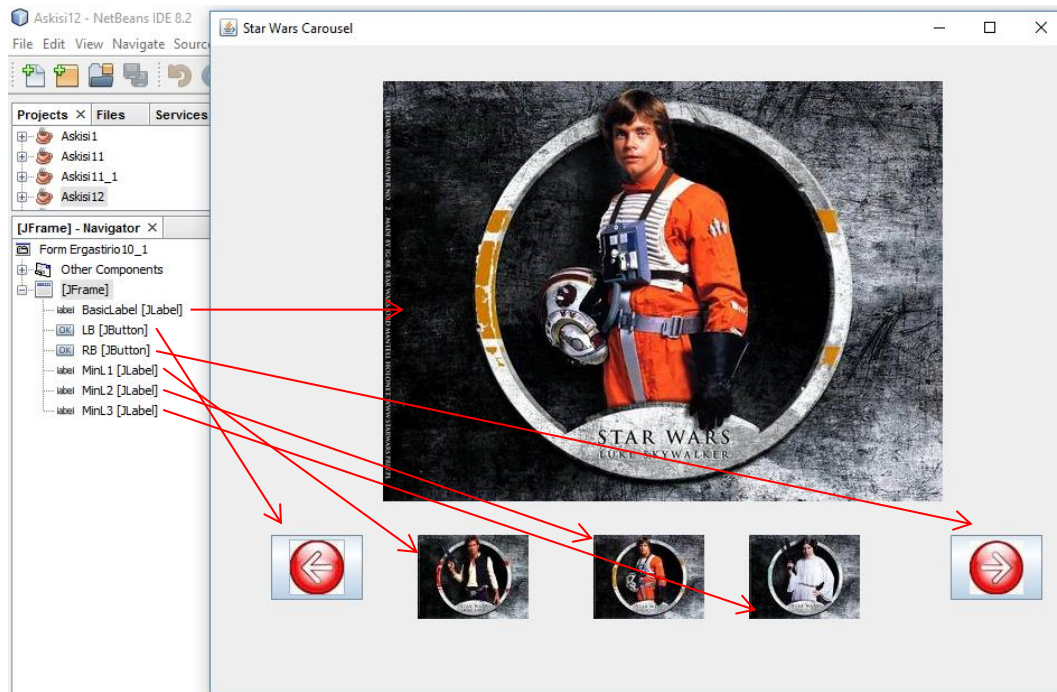
Βήμα 5: Προσθέτουμε τέσσερα JLabel και δύο JButton στον καμβά (JFrame).

Στο πρώτο JLabel θα εμφανίζουμε τη μεγάλη εικόνα. Θα έχει όνομα μεταβλητής BasicLabel και διάσταση 500, 375 (Ιδιότητα Layout, Horizontal Size = 500 και Vertical Size = 375).

Στα άλλα τρία JLabel θα εμφανίζουμε τις κατάλληλες μικρές εικόνες. Θα έχουν όνομα μεταβλητής MinL1, MinL2, MinL3 και διάσταση 99, 75.

Στα δύο JButton θα εμφανίσουμε την εικόνα του αριστερού και δεξιού βέλους. Τα ονόματα μεταβλητών τους είναι LB και RB.

Φροντίζουμε τα συστατικά να τα τοποθετήσουμε στην κατάλληλη θέση (Εικόνα 1).

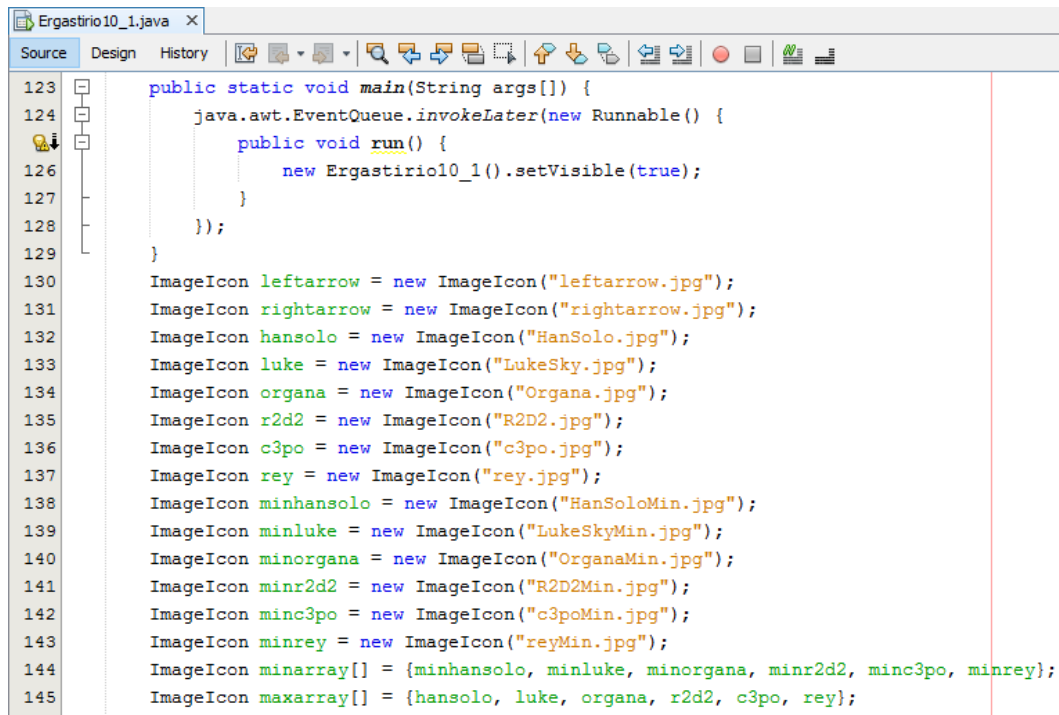


Εικόνα 1: Η σχεδίαση της εφαρμογής

Βήμα 6: Σε αυτό το βήμα ορίζουμε την αρχική εμφάνιση της εφαρμογής μας. Μπορούμε να ορίσουμε τις κατάλληλες ιδιότητες (icon) στα συστατικά της εφαρμογής ή να καλέσουμε τις κατάλληλες εντολές στο δομητή της εφαρμογής αμέσως μετά το initComponents, όπως φαίνεται στην Εικόνα 3. Στην Εικόνα 2, φαίνεται η δήλωση των εικόνων σε αντικείμενα ImageIcon για να μπορέσουμε να τις χρησιμοποιήσουμε στο πρόγραμμα.

Αναλυτικά, στην Εικόνα 2, στις γραμμές 130-143 δηλώνουμε αντικείμενα ImageIcon (έχουμε προσθέσει την κατάλληλη βιβλιοθήκη) που αντιστοιχούν σε εικόνες που έχουμε τοποθετήσει στο Project (στον φάκελο του project). Στη γραμμή 144 δηλώνουμε έναν πίνακα ImageIcon, ο οποίος περιέχει τις έξι μικρές εικόνες σε σειρά και στη γραμμή 145 δηλώνουμε έναν άλλο πίνακα ImageIcon, ο οποίος περιέχει τις έξι μεγάλες εικόνες στην ίδια σειρά.

Στην Εικόνα 3, στις γραμμές 8-13 ορίζουμε τις κατάλληλες εικόνες στα συστατικά της εφαρμογής. Η μεγάλη εικόνα (BasicLabel) θα πρέπει να εμφανίζει την ίδια εικόνα με την μεσαία μικρή εικόνα (MinL2).

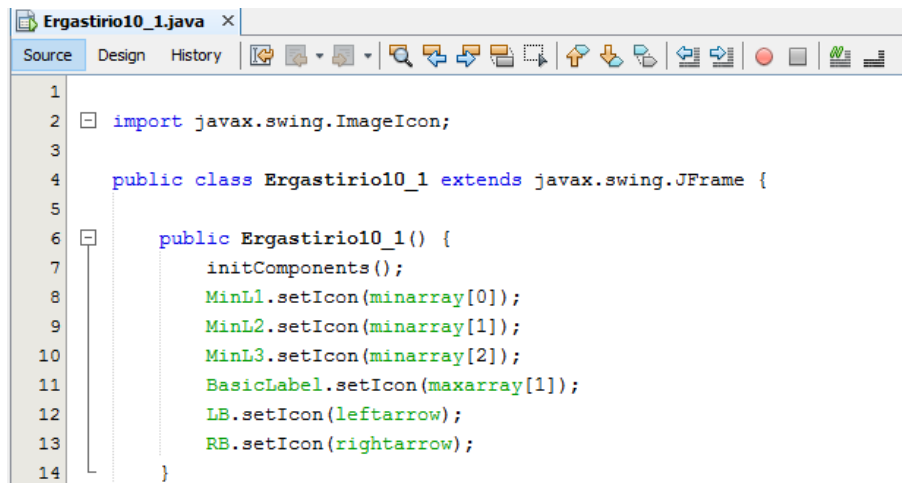


```

123 public static void main(String args[]) {
124     java.awt.EventQueue.invokeLater(new Runnable() {
125         public void run() {
126             new Ergastirio10_1().setVisible(true);
127         }
128     });
129 }
130 ImageIcon leftarrow = new ImageIcon("leftarrow.jpg");
131 ImageIcon rightarrow = new ImageIcon("rightarrow.jpg");
132 ImageIcon hansolo = new ImageIcon("HanSolo.jpg");
133 ImageIcon luke = new ImageIcon("LukeSky.jpg");
134 ImageIcon organa = new ImageIcon("Organa.jpg");
135 ImageIcon r2d2 = new ImageIcon("R2D2.jpg");
136 ImageIcon c3po = new ImageIcon("c3po.jpg");
137 ImageIcon rey = new ImageIcon("rey.jpg");
138 ImageIcon minhansolo = new ImageIcon("HanSoloMin.jpg");
139 ImageIcon minluke = new ImageIcon("LukeSkyMin.jpg");
140 ImageIcon minorgana = new ImageIcon("OrganaMin.jpg");
141 ImageIcon minr2d2 = new ImageIcon("R2D2Min.jpg");
142 ImageIcon minc3po = new ImageIcon("c3poMin.jpg");
143 ImageIcon minrey = new ImageIcon("reyMin.jpg");
144 ImageIcon minarray[] = {minhansolo, minluke, minorgana, minr2d2, minc3po, minrey};
145 ImageIcon maxarray[] = {hansolo, luke, organa, r2d2, c3po, rey};

```

Εικόνα 2: Δήλωση εικόνων σε αντικείμενα ImageIcon



```

1 import javax.swing.ImageIcon;
2
3
4 public class Ergastirio10_1 extends javax.swing.JFrame {
5
6     public Ergastirio10_1() {
7         initComponents();
8         MinL1.setIcon(minarray[0]);
9         MinL2.setIcon(minarray[1]);
10        MinL3.setIcon(minarray[2]);
11        BasicLabel.setIcon(maxarray[1]);
12        LB.setIcon(leftarrow);
13        RB.setIcon(rightarrow);
14    }

```

Εικόνα 3: Εκκίνηση εφαρμογής

Βήμα 7: Σε αυτό το βήμα θα προγραμματίσουμε το JButton LB (Εικόνα 4). Όταν ο χρήστης επιλέγει το «αριστερό βελάκι» γίνεται κύλιση των εικόνων προς τα αριστερά. Άρα, αν στην MinL1 υπάρχει η εικόνα minarray[0], στην MinL2 η εικόνα minarray[1] και στην MinL3 η εικόνα minarray[2] τότε, μόλις ο χρήστης πατήσει το αριστερό βελάκι η MinL1 θα έχει την εικόνα minarray[1], η MinL2 θα έχει την εικόνα minarray[2] και η MinL3 θα έχει την εικόνα minarray[3]. Ουσιαστικά σε κάθε JLabel εμφανίζεται η επόμενη εικόνα από αυτή που εμφανιζόταν.

Αυτό το διαχειριζόμαστε με μία global ακέραια μεταβλητή (index), η οποία αρχικά έχει την τιμή 1. Όταν επιλεγεί το JButton LB αυξάνεται το index κατά 1 (γραμμή 81). Κάθε φορά στην MinL1 θα εμφανίζεται η εικόνα minarray[index - 1], στην MinL2 θα εμφανίζεται η εικόνα minarray[index] και στην MinL3 θα εμφανίζεται η εικόνα minarray[index + 1] και φυσικά στην BasicLabel η εικόνα maxarray[index], όπως εξηγήθηκε στο βήμα 6.

Αυτή η λειτουργία θα δημιουργήσει προβλήματα αν για παράδειγμα το index πάρει την τιμή 6, ενώ ο πίνακας έχει 6 κελιά και επομένως μπορεί να πάρει ως παράμετρο τις τιμές 0 έως 5. Παρόμοια, θα δημιουργηθεί πρόβλημα αν το index+1 αθροίζει σε 6.

Έτσι, στη γραμμή 82 γίνεται έλεγχος της δεύτερης περίπτωσης. Αν το index έχει την τιμή 5 (minarray.length-1) τότε το index+1 θα «βγει» έξω από τα όρια του πίνακα. Σε αυτή την περίπτωση, στις MinL1 και MinL2 θα εμφανιστούν κανονικά οι εικόνες minarray[index - 1] και minarray[index] αντίστοιχα, δηλαδή η προτελευταία και η τελευταία εικόνα του πίνακα minarray. Στην MinL3 δεν μπορεί να εμφανιστεί η εικόνα minarray[index + 1], με άλλα λόγια η minarray[6]. Έτσι, θα εμφανιστεί η επόμενη της τελευταίας, που είναι η πρώτη εικόνα, δηλαδή η minarray[0].

Εικόνα 4: Ο κώδικας του αριστερού κουμπιού

Στη γραμμή 87 γίνεται έλεγχος της πρώτης περίπτωσης. Αν το index έχει την τιμή minarray.length, δηλαδή 6, τότε και το minarray[index] αλλά και το minarray[index + 1] είναι έξω από τα όρια του πίνακα. Έτσι, θα δώσουμε στο index την τιμή 0, η οποία είναι η επόμενη από το 5, αφού θέλουμε όταν «τελειώσουν» οι εικόνες να εμφανίζονται από την αρχή του πίνακα. Επομένως, στην MinL1 θα εμφανιστεί η εικόνα minarray[minarray.length - 1] (δηλαδή

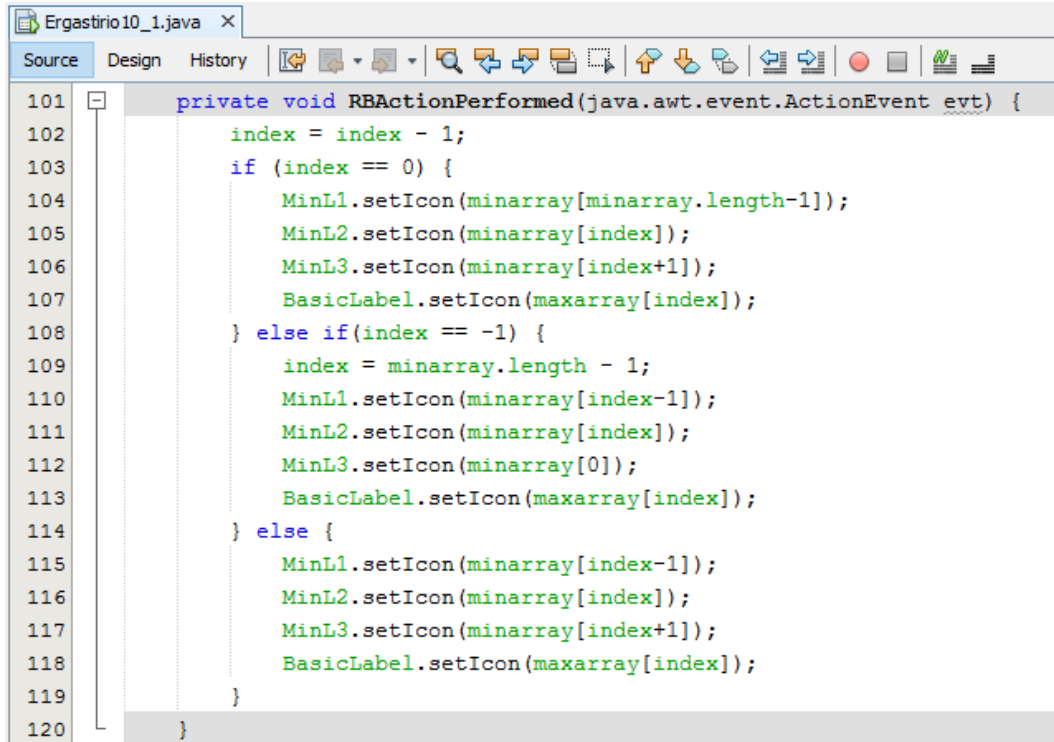
η `minarray[5]`), στην `MinL2` θα εμφανιστεί η εικόνα `minarray[index]` (δηλαδή η `minarray[0]`) και `MinL3` θα εμφανιστεί η εικόνα `minarray[index + 1]` (δηλαδή η `minarray[1]`).

Βήμα 8: Σε αυτό το βήμα θα προγραμματίσουμε το `JButton RB` (Εικόνα 5) με αντίστοιχο τρόπο με το `LB`. Όταν ο χρήστης επιλέγει το «δεξί βελάκι» γίνεται κύλιση των εικόνων προς τα δεξιά. Άρα, αν στην `MinL1` υπάρχει η εικόνα `minarray[1]`, στην `MinL2` η εικόνα `minarray[2]` και στην `MinL3` η εικόνα `minarray[3]` τότε, μόλις ο χρήστης πατήσει το δεξί βελάκι η `MinL1` θα έχει την εικόνα `minarray[0]`, η `MinL2` θα έχει την εικόνα `minarray[1]` και η `MinL3` θα έχει την εικόνα `minarray[2]`. Ουσιαστικά σε κάθε `JLabel` εμφανίζεται η προηγούμενη εικόνα από αυτή που εμφανιζόταν.

Ο προγραμματισμός θα γίνει πάλι με βάση την τιμή του `index`, όπου το `minarray[index]` θα εμφανίζεται στο `MinL2`. Αυτή η λειτουργία θα δημιουργήσει προβλήματα όταν το `index` πάρει την τιμή 0 και -1.

Έτσι, στη γραμμή 103 γίνεται έλεγχος της πρώτης περίπτωσης. Αν το `index` έχει την τιμή 0 τότε το `index-1` θα έχει την τιμή -1 και έτσι θα ζητηθεί εικόνα έξω από τα όρια του πίνακα. Σε αυτή την περίπτωση, στις `MinL2` και `MinL3` θα εμφανιστούν κανονικά οι εικόνες `minarray[index]` και `minarray[index+1]` αντίστοιχα, δηλαδή η πρώτη και η δεύτερη εικόνα του πίνακα `minarray`. Στην `MinL1` δεν μπορεί να εμφανιστεί η εικόνα `minarray[index - 1]`, με άλλα λόγια η `minarray[-1]`. Έτσι, θα εμφανιστεί η προηγούμενη της πρώτης, που είναι η τελευταία εικόνα, δηλαδή η `minarray[minarray.length-1]`.

Στη γραμμή 108 γίνεται έλεγχος της δεύτερης περίπτωσης. Αν το `index` έχει την τιμή -1, τότε και το `minarray[index]` αλλά και το `minarray[index - 1]` είναι έξω από τα όρια του πίνακα. Έτσι, θα δώσουμε στο `index` την τιμή `minarray.length-1`, η οποία είναι η επόμενη το 5, αφού θέλουμε όταν «τελειώσουν» οι εικόνες να εμφανίζονται από το τέλος του πίνακα. Επομένως, στην `MinL1` θα εμφανιστεί η εικόνα `minarray[index - 1]` (δηλαδή η `minarray[4]`), στην `MinL2` θα εμφανιστεί η εικόνα `minarray[index]` (δηλαδή η `minarray[5]`) και `MinL3` θα εμφανιστεί η εικόνα `minarray[0]` (δηλαδή η πρώτη εικόνα η οποία είναι η επόμενη από την τελευταία).



```
101 private void RBActionPerformed(java.awt.event.ActionEvent evt) {  
102     index = index - 1;  
103     if (index == 0) {  
104         MinL1.setIcon(minarray[minarray.length-1]);  
105         MinL2.setIcon(minarray[index]);  
106         MinL3.setIcon(minarray[index+1]);  
107         BasicLabel.setIcon(maxarray[index]);  
108     } else if(index == -1) {  
109         index = minarray.length - 1;  
110         MinL1.setIcon(minarray[index-1]);  
111         MinL2.setIcon(minarray[index]);  
112         MinL3.setIcon(minarray[0]);  
113         BasicLabel.setIcon(maxarray[index]);  
114     } else {  
115         MinL1.setIcon(minarray[index-1]);  
116         MinL2.setIcon(minarray[index]);  
117         MinL3.setIcon(minarray[index+1]);  
118         BasicLabel.setIcon(maxarray[index]);  
119     }  
120 }
```

Εικόνα 5: Ο κώδικας του δεξιού κουμπιού